What Is Claimed Is:

1. A method of generating and managing shared keys for a plurality of members of a cluster, comprising the steps of

(a) system initialization to produce a functionally generated initial shared key;

(b) functional generation of a next shared key; and

(c) key recovery in the event of either compromise or failure of a node.

2. The method of claim 1, wherein step (a) comprises the steps of:

(i) generating a random initial one-time pad $\alpha_{1,2}$ for each member;

(ii) calculating an initial binding parameter θ_1 based on each $\alpha_{1,2}$, where $\theta_1 = \alpha_{1,1} \circledcirc \alpha_{2,1} \circledcirc - - - \circledcirc \alpha_{n,1}$ wherein $\circledcirc$ is a commutative operator; and

(iii) sending θ_1 and $\alpha_{i,1}$ to each member *i*.

3. The method of claim 2, wherein step (iii) comprises the step of encrypting θ_1 and $\alpha_{i,1}$ in the form

$$\left\{ \left\{ T_{SM}, I, 1, \alpha_{i,1} \right\}_{K_{SM}^{-1}} \right\}_{K_i}$$

for transmission to each member *i*, where

T_{SM} is a timestamp generated by a security manager (SM),

I is an indicator of an initialization mode,

1 denotes the first interaction of key generation,

K_{sm}^{-1} is an encryption operation using a private component of a private/public key pair of the security manager, and

K_i indicates encryption using a public component of a private/public bey pair of member i.

4. The method of claim 1, wherein step (a) comprises the steps of:

(i) generation, by a member 1, of random quantities γ and $\nu_{1,1}$;

(ii) calculation by the member 1, of $\gamma \circledcirc \nu_{1,1}=\delta_1$, wherein $\circledcirc$ is a commutative operator;

(iii) sending, by the number 1, of δ_2 to a member 2;

(iv) receipt, by a member i, of δ_{i-1} from a preceding member i-1;

(v) generation, by member i, of random quantity $\nu_{i,1}$;

(vi) calculation, by member i, of $\delta_{i-1} \circledcirc \nu_{i,1}=\delta_i$;

(vii) sending, by member i, of δ_i to a member i+1;

(viii) sending, by a last member n, of δ_n to member 1;

(ix) calculation, by member 1, of $\gamma \circledcirc \delta_n=\theta_1$;

(x) sending, by member 1, of θ_1 to each member;

(xi) calculation, by each member, of $\theta_1 \circledcirc \nu_{i,1} = \alpha_{i,1}$.

5. The method of claim 4, wherein step iii) comprises the step of encrypting δ_1 in the form

$$\left\{\left\{T_1, I, 1, \delta_1\right\}_{K_1^{-1}}\right\}_{K_2}$$ for transmission to member 2,

step (vi) comprises the step of encrypting δ_i in the form

$$\left\{\left\{T_i, I, 1, \delta_i\right\}_{K_i^{-1}}\right\}_{K_{i+1}}$$ for transmission to member i+1,

step (vii) comprises the step of encrypting δ_n in the form

$$\left\{\left\{T_n, I, 1, \delta_n\right\}_{K_n^{-1}}\right\}_{K_1}$$ for transmission to member 1, and

step (ix) comprises the step of encrypting θ_1 in the form $\left\{\left\{T_1, I, 1, \theta_1\right\}_{K_1^{-1}}\right\}_{K_i}$ for transmission to member *i*.

6. The method of claim 1, wherein step (b) comprises the steps of:

(i) generation, by each member *i*, of a cryptographically secure random number, $Fk_{i,j}$, where *j* denotes the key generation iteration;

(ii) calculation, by each member *i*, of $HFK_{i,j} = \alpha_{i,j} \circledcirc FK_{i,j}$, where $\circledcirc$ is a commutative operator;

(iii) sending, by each member *i*, of $HFK_{i,j}$ to each other member;

(iv) calculation, by each member *i*, of

$$\theta_{j+1} = \lambda\theta_j \circledcirc HFK_{1,j} \circledcirc HFK_{2,j} \circledcirc \cdots \cdots HFK_{n,j}$$

where λ is a scaling factor and *n* is the number of members in the cluster;

(v) calculation, by each member *i*, of

$$\alpha_{i,j+1} = \theta_{j+1} \circledcirc FK_{i,j}$$

(vi) calculation, by each member *i*, of a shared key

$$SK_{j+1} = f(\theta_{j+1})$$

where f is a strong one way function, to form a fractionally generated next shared key.

7. The method of claim 6, wherein the step (iii) comprises the step of encrypting $HFK_{i,j}$ in the form $\left\{\left\{T_i, G, j, HFK_{i,j}\right\}_{K_i^{-1}}\right\}_{K_m}$ for transmission to each other member m.

8. The method of claim 6, wherein

step (i) comprises the steps of:

(A) random selection, by each member i, of a number $FK_{i,j}^{-1}$,

where $0 \leq FK_{i,j}^{-1} \leq p\text{-}2$, wherein p is a large odd prime number, such that p-1 has large prime factors; and

(B) calculation, by each member i, of

$$FK_{i,j} = \alpha_{i,j}^{FK_{i,j}^{-1}};$$

step (ii) comprises the step of calculation, by each member i, of $HFK_{i,j} = (\alpha_{i,j} + FK_{i,j}) \bmod p$;

step (iii) comprises the step of encrypting, by each member i, of $HFK_{i,j}$ in the form

$$\left\{\left\{T_i, G, j, HFK_{i,j}, FK_{i,j}\right\}_{FK_{i,j-1}^{-1}}\right\}_{FK_{m,j-1}}$$

for transmission to each other member m;

step iv) comprises the step of calculating, by each member i, of

$$\theta_{j+1} = ((p\text{-}n\text{-}3)\,\theta_j + \sum_{i=1}^{i=n} HFK_{i,j}) \bmod (p\text{-}1)$$

$$= GK_{j+1}^{-1}; \text{ and}$$

step (v) comprises the step of calculation, by each member i, of

$$\alpha_{i,j+1} = (GK_{j+1}^{-1} + FK_{i,j}^{-1}) \bmod p.$$

9. The method of claim 1, wherein step c) comprises the steps of:

(i) sending, by a recovery initiator RI, of the hidden fractional key of a failed node $\bar{i}$, $HFK_{\bar{i},j}$, to a newly elected member i, where j represents the iteration in which node $\bar{i}$ failed;

(ii) sending, by RI, of SK_j to member i;

(iii) performing a distributed initialization process, so that each member l receives a binding parameter ξ and a random pad $\beta_{l,j}$;

(iv) calculation, by each member l, of $HFK_{l,j} = \beta_{l,j} \diamond FK_{l,j}$, where $\diamond$ is a commutative operator;

(v) sending, by each member l, of $HFK_{l,j}$ to member i;

(vi) calculation, by number i, of

$FK_{\bar{l},j} = \lambda\, \xi \diamond HFK_{l,j} \circledcirc \cdots \circledcirc HFK_{n-1,j} \circledcirc \theta_{j+1}$, where $\circledcirc$ is a commutative operator; and

(vii) calculation, by member i, of

$\alpha_{\bar{l},j} = HFK_{\bar{l},j} \circledcirc FK_{\bar{l},j}$

10. The method of claim 9, wherein

step (i) comprises the step of encrypting $HFK_{\bar{l},j}$ in the form

$\left\{\left\{T_{Rl}, R, j, HFK_{\bar{l},j}\right\}_{K_{Rl}^{-1}}\right\}_{Ki}$ for transmission to member i, where R indicates recovery mode;

step (ii) comprises the step of encrypting SK_j in the form

$\left\{\left\{T_{Rl}, R, j, SK_j\right\}_{K_{Rl}^{-1}}\right\}_{Ki}$ for transmission to member i; and

step (v) comprises the step of encrypting $HFK_{l,k}$ in the form

$\left\{\left\{T_l, R, j, HFK_{l,j}\right\}_{K_l^{-1}}\right\}_{Ki}$.

11. The method of claim 2, further comprising the step of

(d) verifying that each of initial pad $\alpha_{i,j}$ has contributed to the calculation of θ_1, performed after step (a).

12. The method of claim 11, wherein step (d) comprises the steps of:

(i) selection, by a predetermined member of the cluster, of a large prime q;

(ii) distribution of q to all members;

(iii) selection, by the predetermined member, of a generator g of the multiplicative group under q;

(iv) distribution of g to all members;

(v) selection by each member i, of a random polynomial f_i having a value of zero at the origin;

(vi) calculation, by each member i, of $\hat{\alpha}_{i,j} = g^{\alpha_{i+1}+f_i}$;

(vii) sending, by each member i, of $\hat{\alpha}_{i,1}$ to all other members;

(viii) calculation, by each member i, of

$$g^{\hat{\theta}_1} = \prod_{j=1}^{j=n} \hat{\alpha}_{i,j} = g^{\theta_1 + \sum_{j=1}^{j=n} f_i}$$

, evaluated at the origin;

(ix) determination, by each member i, of whether $g^{\theta_1} = g^{\hat{\theta}_1}$, evaluated at the origin; and

(x) determination, by each member i, of whether

$$g^{\alpha_{i,j}} = \frac{g^{\theta_1}}{\prod_{j=1}^{j=n} g^{\alpha_{j,1}}}.$$

13. The method of claim 4, further comprising the step of:

(e) verifying that each initial pad $\alpha_{i,1}$ has contributed to the calculation of θ_1, performed after step (a).

14. The method of claim 12, wherein step (e) comprises the steps of:

(i) selection, by a predetermined member of the cluster, of a large prime q,

(ii) distribution of q to all members;

(iii) selection, by the predetermined member, of a generator g of the multiplicative group under q;

(iv) distribution of g to all members;

(v) calculation, by member 1, of g^{γ} and $g^{v_{1,1}}$;

(vi) making g^{γ} and $g^{v_{1,1}}$ available to all members;

(vii) calculation, by each member i, of $g^{v_{i,1}}$;

(viii) publication, by each member i, of $g^{v_{i,1}}$ for other members of the cluster only;

(ix) determination, by each member i, of whether $g^{\theta_i} = \prod_{j=1}^{j=n} g^{v_{i,1}}$.

15. A system for generating and managing shared keys for a plurality of members of a cluster, comprising

initialization means for performing system initialization to produce a fractionally generated initial shared key;

fractional generation means for fractional generation of a next shared key; and

recovery means for performing key recovery in the event of either compromise or failure of a node.

16. A computer program product comprising a computer usable medium having computer readable program code that executes on a computer that participates in the generation and management of shared keys for a plurality of members of a cluster, said computer readable program code comprising:

(a) first computer readable program code logic for causing the computer to participate in system initialization, wherein the initialization produces a fractionally generated initial shared key;

(b) second computer readable program code logic for causing the computer to participate in the fractional generation of a next shared key; and

(c) third computer readable program code logic for causing the computer to participate in key recovery in the event of either compromise of failure of a node.